www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment Report

on

“Predict Disease Outcome Based on Genetic and Clinical Data:Use supervised machine learning to classify patients based on genetic markers, clinical symptoms, and lifestyle factors, predicting whether they are at risk for a particular disease.”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

Name of discipline

By

Harshit Kumar kushwaha

(CSE-AIML-202401100400096)

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Predicting disease outcomes is a critical challenge in modern healthcare, with implications for early diagnosis, personalized treatment, and preventive care. Advances in genetic research and clinical data collection have enabled the development of machine learning (ML) models that can analyze complex datasets to identify individuals at risk of developing specific diseases.

Methodology

- **Data Collection & Preprocessing-**

Genetic Data: Obtain genomic data (e.g., from GWAS studies or sequencing).
Clinical Data: Gather electronic health records (EHRs) including lab results and medical history.
Lifestyle Data: Collect patient-reported surveys on habits (diet, physical activity, etc.).

- **Feature Engineering-**

Genetic Features: Extract relevant SNPs or gene expressions.
Clinical Features: Derive meaningful biomarkers (e.g., BMI, HbA1c levels).

- **Model Evaluation**:

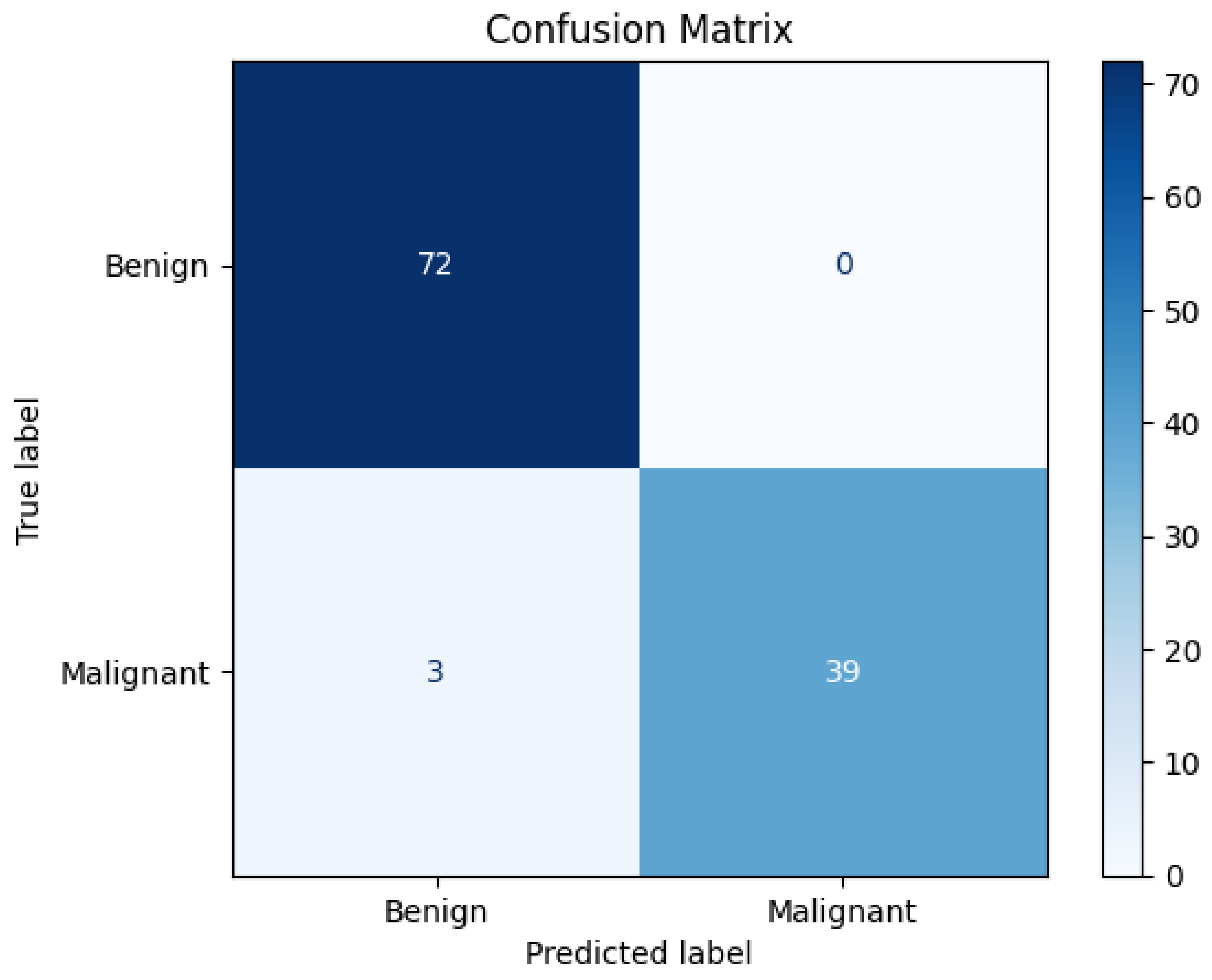
Accuracy: Measures the overall proportion of correct predictions.
Precision: Measures the proportion of true positive predictions out of all positive predictions.
Recall: Measures the proportion of true positive predictions out of all actual positive cases.

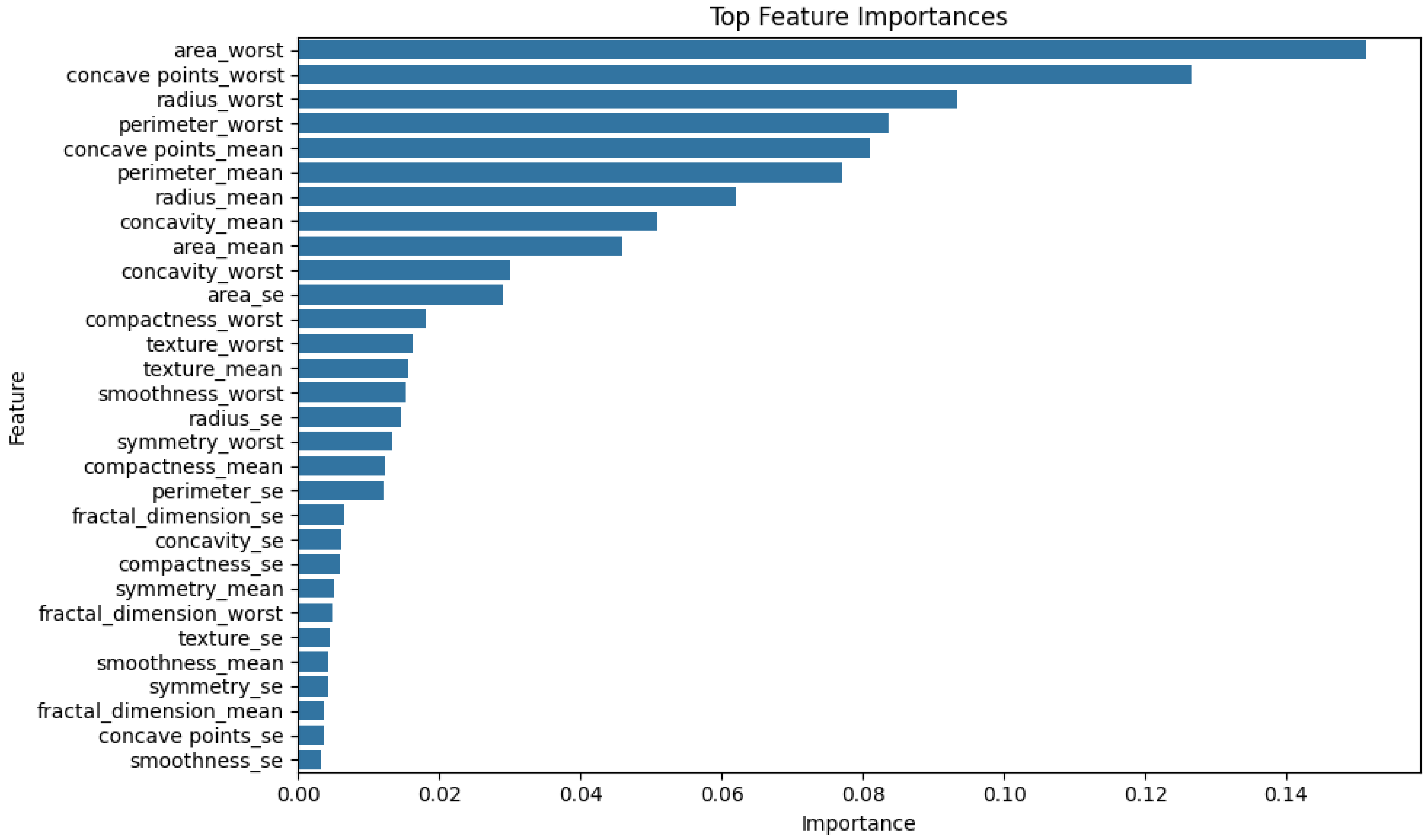
Code

```
# □ Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix, ConfusionMatrixDisplay

# □   Load the dataset
# Make sure the CSV is uploaded to Colab with this exact name
from google.colab import files
uploaded = files.upload()

# Load it into pandas
df = pd.read_csv("3. Predict Disease Outcome Based on Genetic and Clinical Data.csv")

# □   Clean the data
df.drop(columns=["id", "Unnamed: 32"], inplace=True, errors='ignore')
df["diagnosis"] = df["diagnosis"].map({'M': 1, 'B': 0})  # Encode target

# □   Split features and labels
X = df.drop("diagnosis", axis=1)
y = df["diagnosis"]

# □   Train/test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

```
# □  Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# □  Train the model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# □  Predictions and evaluation
y_pred = model.predict(X_test_scaled)

print("□  Accuracy:", accuracy_score(y_test, y_pred))
print("\n□ Classification Report:\n", classification_report(y_test, y_pred))

# □  Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Benign", "Malignant"])
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.grid(False)
plt.show()

# □  Feature Importance
importances = model.feature_importances_
feature_names = X.columns
sorted_idx = np.argsort(importances)[::-1]

plt.figure(figsize=(10, 6))
sns.barplot(x=importances[sorted_idx], y=feature_names[sorted_idx])
plt.title("Top Feature Importances")
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.tight_layout()
plt.show()
```

Output


Confusion Matrix
True label
Benign
Malignant
72
0
3
39
Predicted label
Benign
Malignant
70
60
50
40
30
20
10
0


Top Feature Importances
Feature
area_worst
concave points_worst
radius_worst
perimeter_worst
concave points_mean
perimeter_mean
radius_mean
concavity_mean
area_mean
concavity_worst
area_se
compactness_worst
texture_worst
texture_mean
smoothness_worst
radius_se
symmetry_worst
compactness_mean
perimeter_se
fractal_dimension_se
concavity_se
compactness_se
symmetry_mean
fractal_dimension_worst
texture_se
smoothness_mean
symmetry_se
fractal_dimension_mean
concave points_se
smoothness_se
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
Importance

References

1. For code-chatgpt,deepseek
2. Google Colab